Vysoká škola báňská – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Analýza dat z diskuzních fór

Analysis of Data from Discussion Forums

2010 Tomáš Rozehnal

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 7. května 2010 . . . . . .

ABSTRAKT

V této bakalářské práci se zabývám problematikou návrhu a implementace aplikace pro získání a zpracování dat z diskusních fór na internetu.

V úvodních kapitolách jsou diskutovány hlavní pojmy – sociální síť a diskusní fórum. Následně je popsána jednoduchá analýza internetových fór z pohledu sociálních sítí.

Aby mohla být provedena analýza je nutno z diskusního fóra získat potřebná data. Proto jsou v další části mé bakalářské práce uvedeny postupy a metody pro získávání dat z diskusních fór. Následuje část, pojednávající o zpracování těchto dat. Poté je v práci popsán návrh konkrétní aplikace, kterou jsem udělal pro získávání a zpracování dat.

V poslední části analyzuji získaná data z diskusního fóra. Tato analýza je pouze jednoduchá a lze ji dále rozšířit v rámci následného magisterského studia.

This thesis deals with the problem of the design and implementation for gaining and data processing from internet discussion forums.

In the introductory chapters, the main points are discussed – the social network and the discussion forums.

A simple internet forums´ analysis subsequently follows, from the social networks point of view.

For this analysis to be made, it is necessary to gain the specific dates from the discussion forum. Therefore, the procedures and methods for gaining dates from the discussion forums, and listed in the following parts of my thesis. Further on there is the part dealing with the processing of these dates. Afterwards, the thesis describes the suggestion to a specific application, which I have made to gain and process the dates.

In the last part, I analyze the gained dates from the discussion forum. This analysis is only a simple one; it can be extended furthermore, as part of the Master´s degree course.

KLÍČOVÁ SLOVA

Sociální síť, Diskuzní fórum, Téma (námět), Robot, Parser, XML, GraphML

Keywords

Social network, Discussion forum, Topic, Robot (Crawler), Parser, XML, GraphML

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ASP	– Actvive Server Pages
CIL	– Common Intermediate Language
CLI	– Common Language Infrastructure
CLR	– Common Language Runtime
CLS	– Common Language Specification
CTS	– Common Type Systém
DTD	– Document Type Definition
GraphML	– GraphMarkedLanguage
JIT	– Just-In-Time
LINQ	– Language Integrated Query
MSIL	– Microsoft Intermediate Language
PC	– Personal Computer
Regex, Regexp	– REGular EXpression
URI	– Uniform Resource Identifier
URL	– Uniform Resource Locator
WF	– Workflow Foundation
WCF	– Windows Communications Foundation
WPF	– Windows Presentation Foundation
XML	– eXtensible Markup Language

SEZNAM OBRÁZKŮ

OBSAH:

ÚVOD

Tématem této bakalářské práce je vytvoření a implementace aplikace využitelné v sociálním inženýrství za účelem získání určitých zveřejněných informací. Výstupem této bakalářské práce je software, provádějící získání dat z internetových diskusních fór. Tato aplikace je vhodná při analyzování dat z diskuzních fór z pohledu sociálních sítí. Dále ji lze využít pro snadnější práci s daty, která byla získána z diskusních fór.

Hlavní myšlenkou práce bylo vytvořit jednoduchou a přehlednou aplikaci, ale zároveň použitelnou pro další rozšiřování. Aplikaci s jednoduchým grafickým rozhraním pro uživatele. Aplikaci takovou, jenž je modulární a jednoduše měnitelná pomocí pluginů. Proto je aplikace založená na knihovně, kterou je možné implementovat dále. Tuto knihovnu lze snadno implementovat na jednotlivé internetová fóra. Další knihovna implementuje tento předpis a slouží pro vybírání dat potřebných pouze pro zvolenou analýzu sociální sítě.

V první části mé bakalářské práce jsou rozvedeny jednotlivé pojmy používané v této práci. Jsou zde vysvětlena témata jako sociální síť a diskusní fóra, kterým se popsaná aplikace věnuje.

V následujících částech jsou popsány obecné postupy získání a následné zpracování dat z diskusních fór.

V další části je již podrobněji popsán samotný návrh a implementace aplikace. Následně je uvedeno rozdělení jednotlivých vrstev aplikace, podrobnější popis procházení webu a získávaní dat z diskuzních fór a následné exporty dat do souborů XML a GraphML.

Poslední část této práce je věnována již samotné jednoduché analýze dat získaných pomocí aplikace a zobrazení vygenerované sociální sítě na základě jednoduchého schématu.

V závěru je provedeno shrnutí celé práce, která může sloužit, jako základ pro další rozšíření této aplikace a hlubší analýzy získaných dat v rámci magisterského studia.

1. SOCIÁLNÍ SÍTĚ

Sociální sítě jsou skupiny lidí, které spojuje nějaká vlastnost nebo zájem, na jejímž základě se tito lidé ovlivňují. Síť je možno přirovnat k uzlům, což jsou účastníci. Tyto uzly spojují hrany, což jsou společná témata. Tématy mohou být rodinné vazby, společenské vazby, zájmy atd. Detailně jsou sociální sítě popsány například v [1].

Pojem sociální sítě se dnes často používá ve spojení s internetem a nástupem webů, které se na vytváření sociálních sítí přímo zaměřují. Sociální sítě se mohou vytvářet také v zájmových komunitách kolem určitých webů.

Pro vytváření sociální sítě se využívá mnoha technologií. Mezi technologie pro vytváření sociální sítě patří e-mailové zprávy zasílané lidem, verbální komunikace přes internet – tzv. chat. Můžeme také využít webové aplikace pro sběr dat, nebo vyžít již existující databáze s již vybranými daty. Data se rovněž mohou získávat z dalších bodů zájmů a ukládáním do vhodného souboru. Touto technologií se zabývá tato bakalářská práce. Růst sítě je zajištěn přidávání uzlů do již existující sítě pomocí společných vazeb.

Analýza sociálních sítí se zabývá měřením a mapováním vztahu mezi lidmi, skupinami nebo organizacemi. Jako uzly jsou představeni lidé nebo skupiny a hrany představují informace mezi těmito uzly. Měření sociální sítě se provádí několika metodami.

Základní metody:

- **Těsnost centrality** – zabývá se blízkostí účastníka k ostatním účastníkům v síti.
- **Stupeň centrality** – spočívá v měření aktivity připojení jednotlivých uživatelů.

Velmi důležité vlastnosti pro měření patří:

- Spolehlivost dat
- Chybovost dat
- Správnost dat
- Přesnost dat

1.1 GRAFICKÁ ANALÝZA SOCIÁLNÍCH SÍTÍ:

Tato metoda se převážně využívá v sociálních vědách, protože znázorňuje údaje o komplexních vztazích v určitých skupinách lidí. Sledovaný objekt muže být jednotlivec, organizace

nebo událost, která má vztah s dalšími objekty. Jednotlivé uzly v síti jsou představeny jako body, popřípadě vrcholy a vztahy mezi jednotlivými body jsou znázorněny čarami.

Typy sociálních sítí s využitím orientovaných hran jsou dle [2] následující:

- **Hvězdice** Hlavní uzel se nachází uprostřed pomyslné hvězdy a na něj směřují ostatní odkazy.
- **Řetěz** Jeden uzel ukazuje na druhý ten na třetí atd. Takto tvoří řetěz.
- **Dvojice** Dva uzly na sebe navzájem ukazují
- **Moc** Různě ustavené uzly, jsou mezi sebou různě propojeny. Nakonec tyto uzly končí v jednom uzlu.
- **Izolace** Uzel je samostatný. Není propojen s žádným jiným.

2. DISKUSNÍ FÓRA

Diskusní fórum je elektronické fórum na webových stránkách, ve kterém probíhá on-line diskuse. Obvykle jde o diskusní server, který umožňuje návštěvníkům komunikovat mezi sebou navzájem, ale i se správcem, moderátory nebo vlastníkem webové prezentace.

Fórum slouží uživatelům internetu k svobodné a otevřené diskusi. Jeho dalšími možnostmi je dotazování účastníků na různé věci, nebo naopak pro sdělení nějaké informace okolnímu světu. Diskusní fóra navštěvují stejně smýšlející uživatelé kvůli výměně názorů, zkušeností nebo nabízejí pomoc a odpovědi na různé otázky. Diskutující zde probírají a řeší problémy, které je spojují v jednu komunitu. Uživatelé na fóru získávají cenné rady, jelikož zde mají výhodu komukoliv položit svůj dotaz.

Internetová fóra jsou webové aplikace, které mají zdarma sloužit široké veřejnosti. Obsah fóra je uživatelský generován což znamená, že lidé vkládají své názory a reakce. Ty se následně na stránce zobrazují, čili automaticky generují.

Fórum umožňuje uživatelům zadávat příspěvky do diskuzních témat, o kterých chtějí diskutovat. Dále umožňuje zakládat nová diskusní témata a členit je do skupin. Některá webová fóra povolují tvořit kategorie nebo podfóra. Jednotlivé diskuse jsou vymezeny diskusním tématem. Diskuzní fóra nejsou jenom témata a v nich diskutující uživatelé. Umí toho mnohem více, například upozorňování emailem na nové příspěvky, možnost sdílení obrázků, nastavení různých vzhledů, zasílání soukromých zpráv atd.

Na rozdíl od chatů a IRC kanálů se diskusní fórum liší tím, že přispěvatelé nemusí být připojeni online ke stránce. Tudíž uživatelé nemusejí reagovat okamžitě, ale reagují s časovým odstupem například dní nebo i měsíců. Návštěva a účast na fórech většinou vyžaduje pouze webový prohlížeč a ne další dodatečný software.

2.1 STRUKTURA DISKUSNÍHO FÓRA

Diskusní fóra většinou využívají stromové struktury dat. Tedy příspěvky jsou mezi sebou vázány podle toho kdo, na jaký příspěvek reaguje. Takto strukturované příspěvky tvoří tzv. vlákna. Diskusní fórum je členěno do adresářové struktury. Vychází z hlavní stránky fóra a podle jednotlivých témat se zanořují níže do kategorií případně podfór na konci toho stromu jsou příspěvky v jednotlivých topicích (tématech).

2.1.1 Téma

Témata neboli náměty jsou tvořena jednotlivými příspěvky uživatelů. V komunitě uživatelů se na webu setkáváme se dnes, už vžitým anglickým výrazem topic. Témata tvoří seznam témat diskusních fór. Tyto seznamy témat mohou být ještě rozděleny do jednotlivých kategorií nebo podfór (subfór). Seznam témat je zpravidla zobrazen ve výchozím zobrazení diskusního fóra.

2.1.2 Příspěvky

Jedná se o příspěvky jednotlivých uživatelů. Většinou jsou seřazeny od nejstarších po nejnovější, ovšem toto řazení se dá většinou měnit. Jednotlivé přípěvky obsahují jméno autora, předmět zprávy a následný text příspěvku. Případně se v příspěvcích vyskytují citace, jenž značí na, který příspěvek daný uživatel odpovídá. Dále mohou příspěvky obsahovat bližší informace o autorovi. Na většině webových fór se také měří počet příspěvků jednotlivých uživatelů.

2.1.3 Další struktury aplikace

- **Osobní zprávy -** private messages, soukromé zprávy.

  Jedná se o krátké textové zprávy, které jsou posílány jednomu nebo více registrovaným uživatelům určitého fóra. Zprávu vidí pouze příjemce a odesílatel zprávy.
- **Emotikony: -** smiles, smajlící.

  Jsou symboly nebo kombinace symbolů, které vyjadřují lidské pocity. Diskusní fóra podporují emotikony buď v textové podobě, nebo jsou vloženy jako obrázek.
- **Anketa:**

  Na většině diskusních fór se využívají ankety. Zjišťuje se jimi uživatelská spokojenost. Hlasuje se v nich o případných změnách fóra nebo si jednotlivé ankety mohou tvořit sami uživatele v jednotlivých tématech. Tyto statistiky mohou být buď zobrazeny, nebo mohou být skryty.

2.2 UŽIVATELÉ A UŽIVATELSKÉ SKUPINY

- **Vlastník:**

  Vlastník webové prezentace.

- **Administrátor:**

  Je člověk, který se stará o technické detaily diskusního fóra, jako jsou pravidla fóra, vytváření kategorií, případně témat. Dále také může určovat, který uživatel se stane moderátorem.

- **Moderátor:**

  Jsou to vybraní uživatelé u většího fóra pracovníci, kteří schvalují různá práva uživatelů jako jejich přístupy k příspěvkům. Moderátoru je svěřena část fóra, o kterou se stará a zodpovídá za ni, například má právo mazat příspěvky, neodpovídají-li pravidlům.

- **Registrovaný uživatel:**

Uživatel, který prošel procesem registrace. Obvykle může přispívat i zobrazovat veškeré příspěvky. Registrací je zajištěno, že nikdo jiný nebude pod touto přezdívkou přispívat.

- **Host:**

  Neboli anonym je uživatel, který není přihlášený do diskusního fóra. Tito uživatele většinou mohou jen prohlížet zobrazené příspěvky, ale již nemohou přispívat svými názory. Na fóru nemají žádné práva nebo privilegia.

- **Uživatelské skupiny:**

  Registrovaní uživatelé jsou zařazováni do uživatelských skupin pro lepší organizaci uživatelů. Podle toho v jaké skupině se uživatel nachází, má uživatel větší práva a výsady. Uživatelé mohou být také administrátorem přiřazeni do privilegovaných skupin.

2.3 Řízení a omezování diskusí

Diskusní fóra a jednotlivé diskuse jsou řízeny uživatelskými skupinami. Každý uživatel má určité práva a povinnosti. Na dodržování těchto práv a povinností dohlížejí moderátoři případně administrátoři, kteří mohou jednotlivé příspěvky mazat. Některá fóra dokonce zveřejňují příspěvky až po schválení moderátorem. Dalším omezením může být omezení samotné diskuse.

Diskuse může být:

- Veřejná – zobrazena všem i anonymům.
- Omezená – zobrazena pouze pro čtení nebo přístupná jen některým uživatelům.
- Neveřejná – není administrátorem vůbec zveřejněna

2.4 Etika

Zvyklosti a pravidla jsou na jednotlivých internetových fórech různá. Proto by si je každý uživatel daného fóra měl přečíst a dodržovat. Na dodržování těchto pravidel opět dohlíží moderátoři. Moderátoři mohou cenzurovat příspěvky, které jsou vulgární, které jsou nežádoucí, neboť nejsou k danému tématu nebo jsou dokonce protizákonné. Rovněž jsou mazány příspěvky, které by mohly narušovat zájem zakladatele diskuse.

Tato „cenzura“musí probíhat, aby nedocházelo pohoršování ostatních uživatelů, nebo možnosti vyvolání tzv. flame war, což je zahlcení fóra zbytečnými konfrontacemi mezi názory uživatelů. Takovému zahlcování fóra zbytečnými příspěvky se říká tapetování.
Hlavní pravidla většiny diskusních fór:

- Příspěvek musí být k tématu. Tyto příspěvky se obvykle mažou. Někdy jsou do určité míry trpěny.
- Příspěvek nesmí být vulgární, protizákonný, nesmí napadat jiné uživatele
- Příspěvek nesmí obsahovat reklamu nebo spam, to znamená příspěvky obsahující komerční informace nebo odkazy.

2.5 SOFTWARE

Existuje mnoho různých aplikací pro tvorbu a provozování diskusních fór. Nejčastěji používanými aplikacemi jsou Invision, Power Board a další.

Ovšem nejznámější a nejfrekventovanější aplikací je phpBB. Je to proto, že má uživatelsky přívětivé a přehledné rozhraní a je zdarma. Obsahuje celou řadu nástrojů ke správě fóra, jeho uživatelů, jejich oprávnění aj. bez nutnosti instalace dalších pluginů.

3. ANALÝZA DISKUSNÍCH FÓR Z POHLEDU SOCIÁLNÍCH SÍTÍ

Jak už bylo v předchozích kapitolách napsáno, uživatelé přihlašující se na internetová fóra, jsou lidé, kteří chtějí diskutovat na nějaké téma. Kolem určitého fóra, které má jistou tématiku a přispívající uživatele se rozvijí tzv. virtuální komunity.

Mezi lidmi v těchto komunitách, nebo celých zájmových skupinách na téma jednotlivých diskusí vznikají sociální vazby.

Témata mohou být různá, například počítačové hry, technika, sport, móda, hudba, náboženství nebo politika. To jsou velmi populární oblasti pro jednotlivé diskuse.

Na diskusní fórum se můžeme dívat jako na sociální síť, kterou můžeme analyzovat. Abychom síť mohli analyzovat, potřebujeme mít o diskusním fóru potřebné informace. Proto musíme z fóra získat data pro provedení určité analýzy. Jelikož počet takovýchto dat může být velký, tvoříme z jedné sociální sítě menší sociální sítě. Tyto menší sítě představují pouze část

daného internetového fóra. Tyto části mohou být podfóra nebo kategorie na něž jsou diskusní fóra rozdělena.

Sociální sít je vytvořena na základě uzlů, kterými jsou obvykle uživatelé či autoři jednotlivých příspěvků v jednotlivých tématech. Vzájemnou komunikací mezi těmito uzly vznikají vazby.

Pro provedení analýzy můžeme zvolit různá kriteria. Analýza sociální sít muže být omezena z pohledu časového, tedy můžeme analyzovat pouze časové období, které nás zajímá.

Jednotlivé uživatele můžeme také analyzovat z pohledu počtu příspěvků jako diskutující nebo nediskutující atd.

Ze získaných dat můžeme zpětně zjistit, jak byla síť vytvořena, jací účastnící tvoří její uzly, jaké jsou jejich priority, věk, zájmy atd., s jakou intenzitou se účastní diskuse na dané téma v diskusním fóru apod..

4. POUŽITÉ TECHNOLOGIE V APLIKACI

V následující části dokumentu se nachází základní shrnutí hlavních technologií, použitých pro návrh a implementaci. Vysvětlení .NET technologie je detailněji popsána v [3] literárním pramenu.

4.1 . NET TECHNOLOGIE

Tato platforma byla vyrobena firmou Microsoft, je určená zejména pro aplikace postavené pro prostředí operačního systémů Windows. Jedná se o soubor technologií, které tvoří platformu pro webové, desktopové i Pocket PC aplikace. Množina těchto technologií je nazývána. NET Framework.

Cíle technologie .NET:

- Zjednodušení a urychlení vývoje aplikace.
- Nezávislost na programovacím jazyce, ve kterém je aplikace vyvíjena.
- Společná knihovna tříd.
- Implementace práce s webovými službami, XML soubory.
- Větší bezpečnost

- Správa verzí
- Jednoduchá instalace

Struktura .NET technologie:

- CLI - Common Language Infrastructure. Základní platforma pro specifikaci jádra .NET Frameworku.
- CLR - Common Language Runtime se nachází na nejnižší úrovni. Jedná se o runtime pro .NET Framework.
- MSIL - Microsoft Intermediate Language je intermediárního jazyk, do kterého jsou zkompilovány zdrojové kódy z libovolného programovacího jazyka
- JIT - Just-In-Time kompilátor spouští se v případě, že má být spuštěna aplikace v MSIL kódu.
- CTS - Common Type System. Společný typový systém
- CIL – Common Intermediate Language. Společný jazyk pro .NET Framework
- Base Class Library – Základní knihovna implementovaná nad CLR.
- CLS – Common Language Specification. Integrace více programovacích jazyků

Součásti .NET Frameworku:

- ASP.NET – technologie pro vývoj webových aplikací.
- Windows Workflow Foundation (WF) – technologie pro definování heterogenních sekvenčních procesů.
- Windows Presentation Foundation (WPF) – technologie pro vytváření grafického uživatelského rozhraní aplikací.
- Windows Communications Foundation (WCF) – technologie pro vývoj webových služeb a komunikační infrastruktury aplikací.
- LINQ – Language Integrated Query, integrace objektového přístupu k datům v databázi, XML a objektech.

4.1.1 Namespace System.Text.RegularExpressions

Jedná se o jmenný prostor v .NET Framework pro práci s regulárními výrazy. V tomto jmenném prostoru se již nachází samotný regulární výraz, který je reprezentován třídou Regex. Regulární výraz vytvoříme použitím konstruktoru.

Základní vlastností regulární výrazů, je hledání potřebného řetězce podle vzoru. Pro takovéto hledání slouží metoda Match nebo Matches pro kolekci nalezených řetězců. Prvním argumentem této metody je text, ve kterém se bude vyhledávat druhým je vzor regulárního výrazu pro hledání.

Pro získání hodnoty slouží vlastnost Value. V této práci je ještě využívaná vlastnost Groups, která značí jednotlivé skupiny uložených množin znaků. Jestliže metoda Regex.Match nebo Regex.Matches nic nevyhledá, vlastnost Success bude mít hodnotu False.
Pro nastavení přesnějšího způsobu hledání, se využívá vlastnosti RegexOptions třídy Regex. K ní se přiřazují jednotlivé volby pro vyhledávání. Jejich přehled a popis je uveden v následující tabulce.

Tabulka č.1: Přehled a popis nejpoužívanějších voleb pro hledání.

Compiled	Regulární výraz je přeložen přímo do aplikace. Vyhledávání běží rychleji.
IgnoreCase	Ignoruje velikost písmen.
RightToLeft	Vyhledávání probíhá v textu zprava doleva.
SingleLine	Znak „.“ ignoruje nové řádky.
MultiLine	Výrazy „^“a „$“ upraví na vyhledávání na začátcích a koncích jednotlivých řádků textu.

4.1.2 Namespace System.Threading

Jmenný prostor v .NET Framework, který je určen pro práci s vlákny. Vlákna využíváme pro vytváření aplikací, kde se budou některé operace provádět „paralelně“ tzn. ve více tocích. Operace ovšem nejsou přímo paralelně prováděny, pouze jsou jednotlivým vláknům provádějícím operace přiřazeny časy přístupu k procesoru. Vlákna se takhle na procesoru střídají velkou rychlostí, a proto pozorovateli připadá, že vlákna pracují souběžně.

Nejvyužívanější třídou z tohoto jmenného prostoru je třída Thread, jejíž instance reprezentuje jednotlivá vlákna. Vlákna jsou užitečná v situacích, kdy jsou operace výpočetně složité a na dlouhou dobu by zabíraly výkon procesoru. Uživatel by tak musel čekat na dokončení operace.

V jednoduchých aplikacích, které reagují dostatečně rychle na podněty uživatele, není potřeba využívat vlákna. Naopak využití vláken by nepřineslo vůbec žádný užitek.

V této práci je využíváno pouze jedno vlákno, ale s vlákny jsem počítal kvůli časové náročnosti procházení webu robotem a následnému zpracování webových stránek. Tato časová náročnost se projevila při stahování příspěvků kdy délka stahování a zpracování těchto dat trvala přibližně 20 minut při hloubce zanoření dvě do jednotlivých témat. Proto by se možná pro tuto aplikaci hodilo doimplementovat více vláken. Doporučoval bych cca 10-12 vláken. Tím by se aplikace výrazně urychlila a přitom by tato aplikace nijak výrazně nezatížila server s internetovým fórem.

4.1.3 Namespace System.Net

Tento jmenný prostor se využívá v situacích, kdy je potřeba komunikovat pomocí síťových protokolů s webovými servery nebo jednodušeji se všemi počítači v síti. Komunikace probíhá pomocí standardních objektů, které se v tomto jmenném prostoru nacházejí. V této práci je využit nejjednodušší objekt pro stahování dat z internetu. Jedná se o objekt WebClient. Tomuto objektu se zadává pouze adresa souboru, který se má stáhnout.

Třída WebClient poskytuje velmi vysokou úroveň abstrakce pro stahování nebo nahrávání dat z nebo na webové servery. U této třídy se nezaobíráme detaily, které souvisejí s vlastní komunikací. Třída WebClient pracuje s objekty URI (Uniform Resource Identifier) pro zadání adresy k objektu který se má stáhnout.

Data z webového serveru mohou byt stahována:

a) Rovnou na disk, pomocí metody DownloadFile, která stahuje celé soubory. Dále pomocí metody DownloadData, stahující soubory do bajtového pole. Třetí metodou je DownloadString, kterou jsem využil pro tuto prácí, jelikož tato metoda ukládá data do řetězce. Všechny tyto metody pracují synchronně. Jejich dalšími modifikacemi je asynchronní stahování. V něm se využívají metody DownloadFileAsync, DownloadDataAsync, DownloadStringAsync

b) Data lze získat ze vzdáleného souboru tak, že získáme jeho datový proud. Se kterým se poté pracuje pomocí proudů. Tyto proudy jsou typu Stream ze jmenného prostoru System.IO.

4.2 XML JAZYK

Byl vyvinut a standardizován konsorciem W3C. Jedná se o obecný značkovací jazyk, pomocí kterého se dají snadno vytvářet konkrétní značkovací jazyky pro různé účely. XML slouží také pro široké spektrum různých datových typů. XML dokumenty jsou rovněž využívány jako datový model v nativních XML databázích.

Jazyk XML nevyužívá žádných předdefinovaných značek. Hlavní výhodou je že autor používá vlastní tágy. Pomocí XML jazyku se vytvářejí XML dokumenty, které dodržují přísnou syntaxi, proto tyto tágy musí být párově ukončeny. XML jazyk obsahuje elementy, které jsou identifikovány jménem. Tyto elementy mohou obsahovat další elementy a atributy. Pokud dokument splňuje tuto syntaxi, říkáme, že je správně strukturovaný (well-formed). XML kromě samotného textu v sobě nese i informaci o jeho významu. XML dokumenty vytvářejí tzv. XML strom, kde dané elementy představují listy a atributy jejich vlastnosti.

Validní schéma a strukturu XML dokumentu mohou popisovat jazyky jako DTD nebo XML Schema. V takto definované struktuře se snadno vyhledávají informace, jelikož můžeme mnohem přesněji označit význam informací. Takto strukturovaný dokument mohou posléze zpracovávat různé aplikace obsahující XML parser, který daný dokument načte pomocí předem nadefinovaného rozhraní. Pro vyhledávání v XML dokumentech se používají XML dotazovací jazyky jako XPath nebo XQuery a další.

Další obrovskou výhodou XML dokumentů je jejich snadná transformace na jiný typ XML dokumentu. Pro tuto transformaci se využívá XSLT transformace. Kde pomocí XSL schématu se nadefinuje transformační XML soubor a pomocí něj se dokumenty transformují.

Příklad č. 1: Struktura správně strukturovaného XML dokumentu:

```
<?xml version="1.0" encoding="windows-1250"?>
<dokument>
  <element>text</element>
  <element atribut = "hodnota">text</element>
</dokument>
```

4.3 GRAPHML JAZYK

Jedná se o formát souboru speciálně vytvořený pro popis grafu. Je jednoduchý na používání a je komplexně použitelný pro popis vlastností grafu. GraphML soubor je založen na formátu XML. Skládá se z jádra jazyka pro strukturovaný popis hlavních vlastností a dalších vlastností, které jsou specifické pro jednotlivé aplikace využívající tento typ souboru.

GraphML soubory [3] [4] podporují různé typy grafů ať již orientovaných, neorientovaných, hyperbolických grafů a mnoho dalších typů nebo jak přesně specifikované grafy pro jednotlivé aplikace. Velkou výhodou je, že GraphML je snadno parserovatelný jelikož je založen na XML dokumentu. Díky tomuto dobrému strukturovanému dokumentu se GraphML soubor využívá pro archivaci, zpracování nebo generování grafů.

Syntaxe je popisována pomocí popisovacího jazyku GraphMLSchema, kde se dodržuje přísná syntaxe, ale také může být popsána pomocí DTD (Document Type Definition).
GraphML soubory jsou zpracovány tak, že jednotlivé aplikace si určují s jakými prvky pracovat a které ignorovat.

Příklad č. 2: Struktura GraphML souboru:

```
<?xml version="1.0" encoding="UTF-8">
<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
  http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
        <graph id="Graphic" edgedefault="undirected">
          <node id="node1"/>
          <node id="node2"/>
          <edge id="edge1" source="node1" target="node2"/>
        </graph>
</graphml>
```

4.4 REGULÁRNÍ VÝRAZY

Podrobně jsou regulární výrazy popsány v [5]. Regulární výraz představuje určitý vzor pro textové řetězce. Jedná se o speciální řetězec znaků, který tvoří masku, pomocí které můžeme v řetězci vyhledávat nebo provádět korekci řetězce. Například kontrolovat data zadávané uživateli. Pomocí regulárních výrazů můžeme řetězce, texty nebo dokonce celé dokumenty generovat.

Regulární výrazy mají velký počet zástupných znaků, ať již pro jednotlivé znaky nebo pro celé množiny znaků i pro přesné sekvence znaků. Můžeme určovat, které znaky nebo množiny znaků se mohou nebo naopak se nesmějí vyskytovat. U regulárních znaků můžeme, také využívat kvantifikátory. Kvantifikátory určují, kolikrát se smí znak opakovat. Dále mohou být ještě tzv. nenasytné nebo líné. Regulárními výrazy můžeme také vybírat hranice řetězce, jelikož mají zástupné znaky pro začátek a konec řetězce.

Jednoduché regulární výrazy obstarají mnoho práce. Můžeme je otestovat v on-line testerech regulárních výrazů. Pro vytváření složitějších výrazů se hodí programy pro tvorbu regulárních výrazů.

Regulární výrazy v současné době podporují téměř všechny programovací jazyky. Většina novějších programovacích jazyků používá regulární výrazy odvozené od jazyka Perl.

V této práci jsem využil klasický nedeterministický algoritmus pro zjišťování, zda v prohledávaném textu se nachází řetězec, který vyhovuje zadanému regulárnímu výrazu.
Při hledání vzoru algoritmus využívá rekurzi. Algoritmus jde od začátku řetězce. Jakmile najde první vyhovující řetězec tak hledání ukončí. Poté vezme celou zbývající část řetězce a hledá znak odpovídající konci regulárního výrazu s tím rozdílem, že se posouvá opačně od konce řetězce. Tento algoritmus jsem využíval pro skupiny znaků mezi jednotlivými tágy na webové stránce.

Příklad č. 3: Regulární výraz pro hledání znaků v jednotlivých řádcích:

```
<tr>.*</tr>
```

Při tomto vyhledávání se objevila vlastnost regulárních výrazů, která se nazývá nenasytnost. Regulární výraz se snažil co nejvíce roztáhnout, vzal tedy první tág <tr> a poslední tág </tr> a vybral vše mezi těmito tágy. Já jsem však potřeboval získat text uzavřený právě jedním párem tágů. Proto jsem využil líného kvantifikátoru „?“. Ten z nenasytného kvantifikátoru „*“ udělá kvantifikátor líný a snaží se najít co nejkratší vyhovující řetězec.

Dále jsem využil speciální konstrukce pro zapamatování toho nalezeného řetězce, pro další práci s tímto řetězcem. Použití je jednoduché, část, kterou si potřebujeme zapamatovat, uzavřeme do kulatých závorek a později se na ni odkážeme. Odkazoval jsem se na tuto skupinu znaků pomocí Regex.Match.Groups[i], kde „i“ představovalo hloubku zapamatované skupiny znaků.

Příklad č. 4: Hledání znaků v jednotlivých řádcích s využitím líného kvantifikátoru a následné vybrání řetězce:

Vybrání jednotlivého řádku: `<tr>.*?</tr>`

Vybrání řádku se zapamatováním: `<tr>(.*?)</tr>`

Výběr regulárního výrazu v programu:

```
MatchCollection rows = Regex.Matches
                       (strContent, "<tr>(.*?)</tr>");
...
string row = rows[i].Groups[1].Value;
...
```

5. POSTUPY A METODY PRO ZÍSKÁNÍ DAT

Nezbytností pro analýzu sociálních sítí jsou data. Potřebná data jsou na jednotlivých internetových fórech uložená v jednotlivých tématech. Proto je nutnost se dostat právě na tyto témata.

5.1 PROCHÁZENÍ DISKUSNÍM FÓREM

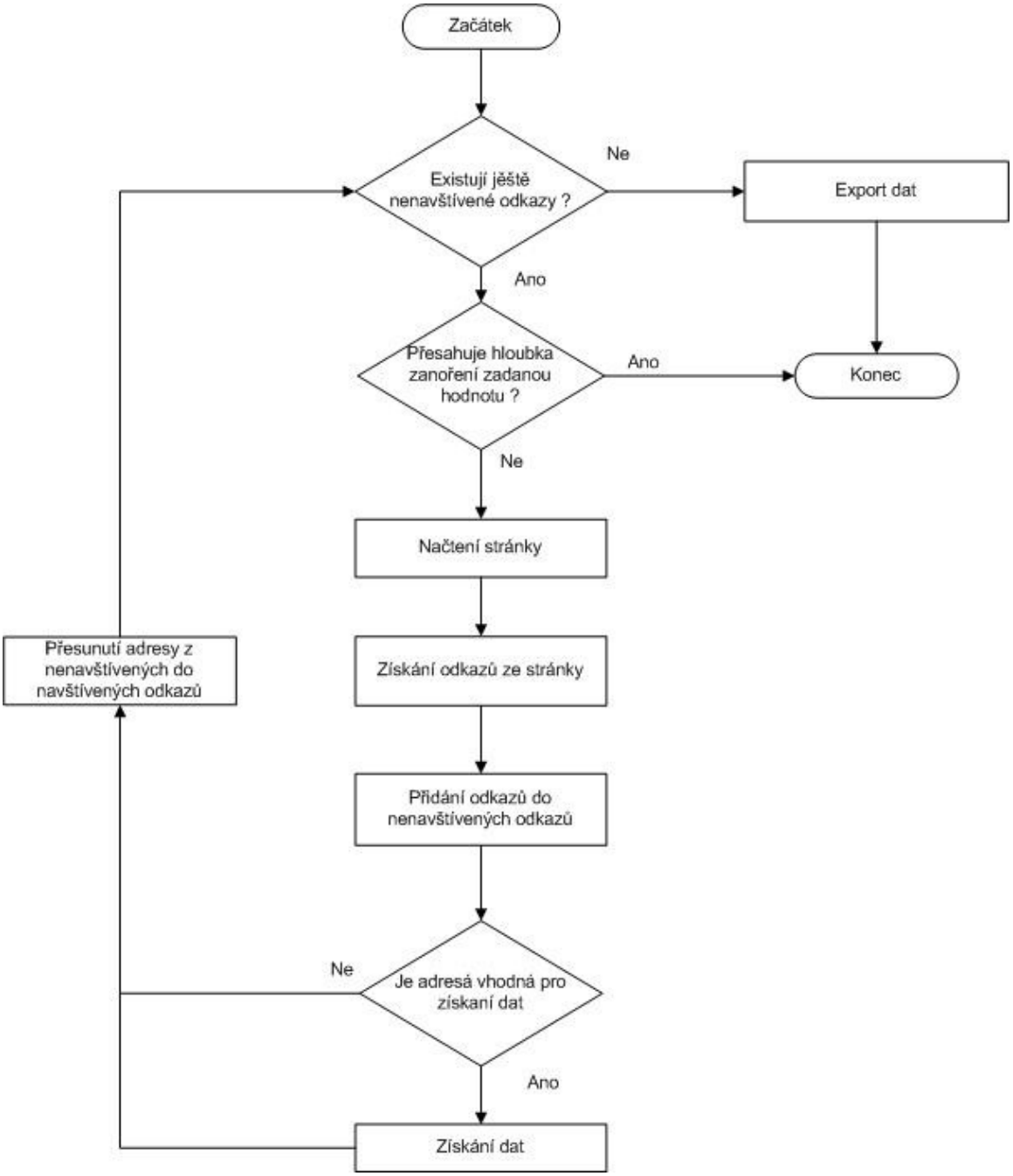
Pro procházení webu, jsem vytvořil robota, který prochází celé diskuzní fórum a vybírá z něj potřebné informace. Na všech stránkách vybere odkazy na další stránky webu a případně odpovídá-li stránka vzoru vyparseruje z ní také potřebná data. Potřebná data v této práci jsou data identifikující uživatele nebo jednotlivé příspěvky na fóru.

Při tvorbě robota jsem vycházel z [6] a vlastností webových prezentací, které mají strukturu stromu, která vychází z domovské stránky tzv. Homepage a větví se dále. Proto jsem nejdříve inicializoval tuto domovskou stránku, ze které robot vychází a posílá ji parseru pro další zpracování.

Pro správné procházení jsem potřeboval vytvořit dvě datové struktury pro ukládání již zpracovaných stránek a stránek nezpracovaných. Nezpracované odkazy jsou určeny pro robota, který je musí projít. Zpracované odkazy slouží pro kontrolu, kde již robot byl, ať nevznikají duplicity jednotlivých stránek i dále zpracovávaných dat. Robot po zpracování stránky parserem dat, odkaz přemístí z nezpracovaných odkazů do zpracovaných odkazů. A takto pokračuje do konečného zpracování všech URL stránek v nenavštívených odkazech. Poté práce crawleru končí.

Robot při procházení stránek pracuje s hloubkou zanoření stránky k hloubce, jež je zadána uživatelem v grafickém rozhraní. Přesahuje-li hloubka zanoření zadanou hloubku, tento odkaz se do seznamu nenavštívených adres nepřidá. Práce robota je zobrazena na následujícím vývojovém diagramu – viz obr.číslo 1.

Obrázek č. 1: Zjednodušené schéma procházení internetovým fórem



5.2 STAŽENÍ WEBOVÉ STRÁNKY

Pro procházení webového fóra potřebujeme dostat data z webové stránky. Proto se na tuto stránku musíme dostat. Pro jednotlivé webové stránky je vytvořena vlastní třída. Do této třídy se ukládá webová adresa jako objekt Uri s dalšími vlastnostmi.

Stažení zdrojového kódu webové stránky probíhá pomocí stažení stringu dané webové stránky. Pro tuhle možnost stahování jsem se rozhodl poté, co jsem pro stahování používal asynchronní stahování stránky do složky Temp. Po parserování stránky jsem tyto data musel následně mazat. Proto jsem se rozhodl pro metodu WebClient.DownloadString(Uri).

Do třídy se strukturou webové stránky se ještě ukládá kódování této webové stránky. Pro jednoduší následnou práci s řetězcem webové stránky.

Jestliže je webová stránka bez problému stažená, je vrácena robotu, který s danou stránkou dále pracuje.

5.3 VYBÍRÁNÍ DAT

Pro výběr dat jsem využil regulárních výrazů v knihovně RegularExpressions. Rozhodl jsem se tak poté, co jsem zkoušel využít XSLT transformaci mezi soubory formátu XML. Chtěl jsem využít toho, že webová stránka je vlastně XML dokument a pomocí transformace bych z něj udělal jiný XML dokument. Z tohoto vlastního dokumentu bych mohl jednoduše vybírat data jako jednotlivé uzly dokumentu nebo jeho atributy. Jenže problém nastal při XSL transformaci, když webové stránka měla špatně ukončené tágy, nebo atributy nebyly v uvozovkách a byla vyvolaná výjimka. Proto jsem musel využít složitých, ale všemocných regulárních výrazů.

5.3.1 Vybírání odkazů

Pro vybírání odkazů jsem vytvořil vzor, znamenající odkaz z těla webové stránky. Pro zjištění zda se tento vzor nachází na webové stránce, jsem využil kolekci Regex.Matches. Procházel jsem jednotlivé výsledky tohoto vzoru a na nich prováděl následné oříznutí řetězce, značící odkaz „a href´=“. Dostal jsem tedy odkaz směřující dále do diskusního fóra. Tyto odkazy byly následně vyčištěny od zbytečných stránek. Dále například od zbytečných session id, které identifikují

uživatele pro jednotlivá sezení na webových stránkách. Nebo jsem nahrazoval stránky seřazené sestupně, řetězce „desc“ na „asc“, ať nevznikají duplicity dat a aby se robot dostal na všechny webové stránky. Po těchto úpravách odkazů jsem k nim přidal kořenovou adresu diskusního fóra. A takto celý odkaz připravený pro následné stáhnutí tohoto odkazu byl uložen do listu s odkazy, který jsem následně vrátil robotovi. Robot s těmito vyparserovanými odkazy pracoval dále. Pro toto parserování jsem vytvořil dvě třídy. Jednu pro parserování odkazů pouze s uživatelskými daty. Druhou pro parserování odkazů pro procházení jednotlivých podfór a témat.

5.3.2 Vybírání dat pro analýzu

Při vybírání dat jsem se musel zaměřit na přesnou strukturu webu. Pro správnou funkčnost jsem implementoval dva různé parsery.

První je parser pro uživatelská data. Tento parser nejdříve v řetězci s webovou stránkou nahradil bílé znaky znakem „$“. Takto jsem vytvořil jednořádkový řetězec, ze kterého jsem vybíral jednotlivé řádky mezi tágy <tr> a </tr>. Z těchto řádků webové stránky jsem dále vyparseroval sloupce. Sloupce jsou na webových stránkách mezi tágy <td> a </td> takto jsem vybral jednotlivé sloupce. Na sloupcích jsem využil vybírání řetězce pomocí Regex.Match.Groups, který vybírá data z uložené skupiny znaků mezi zadanými regulárními výrazy. Následně jsem znak „$“ opět nahradil bílými znaky. Tyto data byly následně ukládány do struktury dat s uživatelskými daty a exportovány do již připravené struktury XML souboru.

Druhý parser slouží pro vybírání dat z jednotlivých témat. Tento parser nepřevádí řetězec stránky na jednořádkový řetězec, protože struktura webové stránky není identická, ale jednotlivé data pro vybrání jsou zakomponována u klíčových slov. Na základě těchto klíčových slov jsem sestavoval regulární výrazy. Data z regulárních výrazů jsem vybíral pomocí Regex.Match.Groups to v případě, že se jednalo o jedinečná data, například název daného tématu. Nebo pomocí Regex.Matches.Groups v případě různých hodnot dat, například jména autorů jednotlivých příspěvků. Takto získaná data parser ukládal do datových struktur a následně exportoval do zvolených souborů.

6. POSTUPY A METODY PRO ZPRACOVÁNÍ DAT

Pro zpracování jednotlivých dat jsem jako mezikrok vytvořil datové struktury. Z těchto datových struktur probíhalo následné exportování dat do XML souborů s daty uživatelů a daty s příspěvky.

6.1 DATOVÉ STRUKTURY

Aplikace obsahuje dvě datové struktury. Jednu strukturu pro data identifikující uživatele a druhou pro jednotlivé příspěvky.

Struktura s daty o uživatelích uchovává tyto data:

- Nick uživatele.
- E-mailovou adresu uživatele (je-li vyplněna).
- Adresu bydliště uživatele (je-li vyplněna).
- Webovou adresu uživatele (je-li vyplněna).
- Datum registrace uživatele.
- Počet příspěvků uživatele.

Struktura s jednotlivými příspěvky uchovává tyto data:

- Název podfóra.
- Název tématu.
- Nick autora příspěvku.
- Předmět příspěvku (je-li uveden).
- Datum vložení příspěvku.
- Text příspěvku.

6.2. EXPORT VYBRANÝCH DAT

Exporty dat jsou popsány například v [7].

6.2.1 Export do XML souboru

Jednotlivé datové struktury jsou exportovány do zvláštních XML dokumentů. Cesty kde, mají být tyto XML dokumenty uloženy, si uživatel vybere v uživatelském rozhraní aplikace.

Na základě poznatků v [7] jsem vytvořil příklady struktur XML dokumentů:

Příklad č. 5: Struktura XML dokumentu pro uživatelská data:

```
<?xml version="1.0" ?>
<Users>
      <User>
      <Name>Nick uživatele</Name>
      <Email>E-mailová adresa uživatele</Email>
      <Registration>Datum registrace</Registration>
      <Address>Adresa bydliště uživatele<Address />
      <Posts>Počet příspěvků uživatele</Posts>
      <Web>Webová adresa uživatele<Web/>
      </User>
</Users>
```

Příklad č. 6: Struktury XML dokumentu pro jednotlivé příspěvky:

```
<?xml version="1.0"?>
<Data>
      <Post>
      <Forum>Název podfóra</Forum>
      <Topic>Název tématu</Topic>
      <Author>Nick autora</Author>
      <Title>Předmět příspěvku</Title>
      <Time>Čas vložení příspěvku</Time>
      <Text>Text příspěvku</Text>
      </Post>
</Data>
```

6.2.2 Export do GraphML souboru

Z datové struktury příspěvků jsou vybrány data do GraphML souboru ve tvaru. Kdy jednotlivé hrany jsou autoři příspěvků a hrany mezi těmito uživateli jsou jednotlivé témata.

Na základě poznatků v [3][č] jsem vytvořil příklady struktur XML dokumentů:

Příklad č. 7: Struktura vygenerovaného GraphML souboru:

```
<?xml version="1.0" encoding="utf-8"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns">
  <graph edgedefault="undirected">
    <key id="name" for="node" attr.name="Uživatelské
     jméno" attr.type="string" />
    <node id="Autor1" />
    <node id="Autor2" />
    <node id="Autor3" />
    <node id="Autor4" />
    <edge source="Autor1" target="Autor2" />
    <edge source="Autor1" target="Autor3" />
    <edge source="Autor1" target="Autor4" />
    <edge source="Autor2" target="Autor3" />
  </graph>
</graphml>
```

7. NÁVRH APLIKACE PRO ZÍSKÁNÍ A ZPRACOVÁNÍ DAT

Aplikaci pro získání dat z diskusních fór jsem sestrojil v .NET Frameworku konkrétně v jazyce C# (vyslovuje se See Sharp).

Navrženou aplikaci jsem strukturoval do vrstev, které se nacházejí v jednotlivých modulech. To zaručuje přehlednost a jednoduchost této aplikace. Jednotlivé moduly neboli části aplikace, jsou jednoduše nahraditelné nebo jinak využitelné. Jednotlivé vrstvy aplikace představují jednoduché uživatelské rozhraní, jádro aplikace, a logiku aplikace.

7.1 JÁDRO APLIKACE

Jádro aplikace představuje jednotlivé jmenné prostory, nacházející se v hlavním jmenném prostoru, který jsem nazval Crawler. Tyto jmenné prostory jsou popsány níže. Obsahují pomocné třídy, vlastnosti a rozhraní starající se o procházení a získávání dat z jednotlivých webových stránek. Jedná se o jednoduché předpisy, na kterých jsem poté implementoval část logiky aplikace.

7.1.1 Namespace Download

Tento jmenný prostor popisuje stahovač neboli downloader stránek a jeho vlastnosti. Ve jmenném prostoru se nachází třída DownloaderProxy, která uchovává informace o nastavení proxy serveru je-li využíván. Nachází se zde také třída udržující přihlašovací údaje uživatele, jeli autentizace na server nutná. Další částí je rozhraní IPage, které popisuje staženou webovou stránku. V tomto rozhraní se nachází vlastnosti, určující zdrojovou adresou, kódování stránky a její obsah. Dalším rozhraním je IPageDownloader, jenž je předpis pro samotné stahování dat z internetu, s vlastnostmi s nastavením proxy serveru a přihlášení uživatele a metodou, která vrací staženou webovou stránku.

7.1.2 Namespace Analyzing

Jmenný prostor, ve kterém se nachází popisující parser dat. Toto rozhraní je pojmenováno IPageParser. Tento interface obsahuje předpis pro metodu ParseLinks, která robotovi vrací odkazy nacházející se na webové stránce. Dále obsahuje předpis metody ParseData, která provádí parserování jednotlivých dat z webové stránky.

7.1.3 Namespace Crawling

Je jmenný prostor pro procházení webovými stránkami a posílá je parseru pro další zpracování. V tomto rozhraní jsou naimplementovány třídy s argumenty pro události. První třídou jsou argumenty pro událost, která indikuje kompletní projití celého serveru. Dalšími třídou jsou

argumenty pro událost, která je vyvolána při začátku stahování nové stránky. Dále se v tomto jmenném prostoru nachází rozhraní ICrawler, které popisuje samotného robota pro procházení diskusního serveru. Obsahuje deklarace jednotlivých událostí s argumenty uvedenými výše a dalšími vlastnostmi pro nastavení stahovače a parseru webových stránek. V tomto rozhraní jsou nadeklarovány metody pro odstartování nebo zastavení procházení webu.

7.2 UŽIVATELSKÉ ROZHRANÍ

V uživatelském rozhraní jsem připravil hlavní formulář, se kterým pracuje uživatel. Formulář se nachází ve třídě General. Zde si uživatel aplikace zvolí diskusní fórum, ze kterého chce stáhnout data. Vybere si hloubku, do jaké se mají data v jednotlivých tématech stahovat. Uživatel na tomto formuláři také vybírá jednotlivá úložiště dat, kam chce ukládat jednotlivé soubory s příspěvky, daty a GraphML soubor.

Volby popsané výše musí být uživatelem nastaveny, jinak je uživatel při kliknutí na tlačítko zajišťující stahování upozorněn, že některá volba není vybrána. Jsou-li tyto volby zadány, nastaví vytvořená aplikace jednotlivé atributy a vlastnosti a spustí stahování. Uživatel si zde také může zadat přihlašovací údaje k tomuto diskusnímu fóru. Po začátku stahování může uživatel toto stahování zastavit po kliknutí na tlačítko „Zastavit“. Toto tlačítko zobrazí další okno, kde uživatel tuto volbu potvrdí. Na tomto formuláři je také možnost práce s horním menu aplikace, kde uživatel může zadat nastavení proxy serveru nebo zobrazit informace o aplikaci.

Uživatelské rozhraní obsahuje i tzv. splash obrazovku, která se zobrazí při načítání aplikace. Dalším formulářem je již zmíněné nastavení proxy serveru, které obsahuje validaci vstupů. Posledním formulářem je formulář, kde uživatel zjístí základní informace o aplikaci.

7.3 IMPLEMENTACE JÁDRA APLIKACE

Implementace jádra je nejdůležitější částí aplikace, kterou jsem vytvořil. Rozdělil jsme ji do jednotlivých tříd.

Třída představující webovou stránku je nazvána DefaultPage, která provádí implementaci rozhraní IPage.

Další třídou je DefaultUrlManager, která se stará o řízení jednotlivých URL adres. Tato třída obsahuje dvě datové strukturu typu Dictionary. Jedna struktura slouží pro ukládání

navštívených stránek a druhá pro ukládaní nenavštívených stránek. Tato třída se stará o jednotlivé vkládání do těchto struktur. Kontroluje, jestli se do těchto struktur může vkládat, jestli už není daná URL vložena. Dále tato třída obsahuje metodu, která předává robotu jednotlivé nenavštívené adresy.

Další částí implementace jádra je stahovač stránek. Nachází ve třídě DefaultPageDownloader. Stahovač implementuje samotné stahování webových stránek. Obsahuje také vlastnosti proxy a přihlášení uživatele.

Tato část aplikace implementuje dva různé parsery. První parser se stará o parserování odkazů a dat jednotlivých uživatelů. Tento parser se nazývá MemberListParser a implementuje metody pro parserování uživatelských dat. Druhý parser se jmenuje DefaultParser a implementuje metody pro jednotlivé příspěvky.

Dalšími implementacemi jádra jsou jednotliví roboti. První je implementován pro procházení stránek s uživatelskými informacemi. A druhý pro procházení stránek s jednotlivými příspěvky. Tyto implementace jsou téměř identické, až na to, že robot pro procházení stránek s uživatelskými informacemi si sám počítá hloubku zanoření, aby vybral všechny uživatele. Robot pro procházení stránek s jednotlivými příspěvky dostává tuto hloubku zadánu uživatelem. Oba roboti implementují rozhraní ICrawler. Hlavní metodou těchto robotů je metoda CrawlerTreadProcedure, která vytvoří kořenovou stránku fóra. Inicializuje UrlManager , kterému předá tuto stránku. Poté pracuje v cyklu, dokud v UrlManageru existují nenavštívené adresy. V tomto cyklu vezme nenavštívenou adresu z UrlManageru a pomocí ji Downloaderu stáhne. Jestliže tato adresa není prázdná, předá se parseru, který parseruje data a následně odkazy. Pokud odkazy nejsou z jiné domény, robot jim zvětší vlastnost znamenající hloubku zanoření. Tyto odkazy jsou vkládány do nenavštívených adres UrlManageru. Jestliže se v UrlManageru nenacházejí žádné již nenavštívené adresy, práce robota skončí.

7.5 DATOVÉ STRUKTURY

Popis datových struktur, které jsem v aplikaci vytvořil jsou uvedeny v kapitole 6, a proto je již znovu nebudu popisovat.

7.6 EXPORT DATOVÝCH STRUKTUR

Poslední částí mojí aplikace je export datových struktur.

7.6.1 Export příspěvků a dat o uživatelích do XML souboru

Při těchto exportech jsem využil třídu XmlDocument, která představuje objektový model pro XML data tzv. DOM. V XML dokumentu je na začátku nastavena deklarace. Poté je nastaven kořenový element, do kterého jsou zanořeny jednotlivé XML Elementy představující příspěvky nebo uživatele. Poté je kořenový element ukončen a celý tento XML dokument uložen do jména souboru, který si uživatel vybral.

7.6.2 Export dat z příspěvků do GraphML souboru

Ve třídě, určené pro export dat do GraphML souboru, je nadeklamována datová struktura Dictionary, která obsahuje jednotlivé témata a autory, kteří do nich přispěli. Tato třída obsahuje metodu, která vkládá autory do jednotlivých témat.

Další metoda tyto autory poté spojuje v dlouhý seznam autorů, který poté využívá metoda, která generuje GraphML soubor. Metoda pro generování GraphML souboru využívá třídu XmlTextWriter, které nastaví cestu pro uložení souboru a kódování. Při generování toho souboru je nastavena deklarace různých uzlů a atributu toho dokumentu. Poté se do dokumentu zapisuji jednotlivé uzly a hrany grafu. Po vložení těchto uzlů a hran je dokument ukončen a uzavřen.

8. ANALÝZA DAT Z DISKUZNÍCH FÓR

Jednoduchou analýzu získaných dat jsem prováděl pomocí programu Microsoft NodeXL[6]. Microsoft NodeXL je rozšíření pro aplikaci Microsoft Office Excel. Jedná se o kvalitní nástroj pro pokročilou vizualizaci a analýzu sociálních sítí. Program je jednoduchý na užívání, a pracuje s tabulkovým rozhraním Microsoft Office Excel.

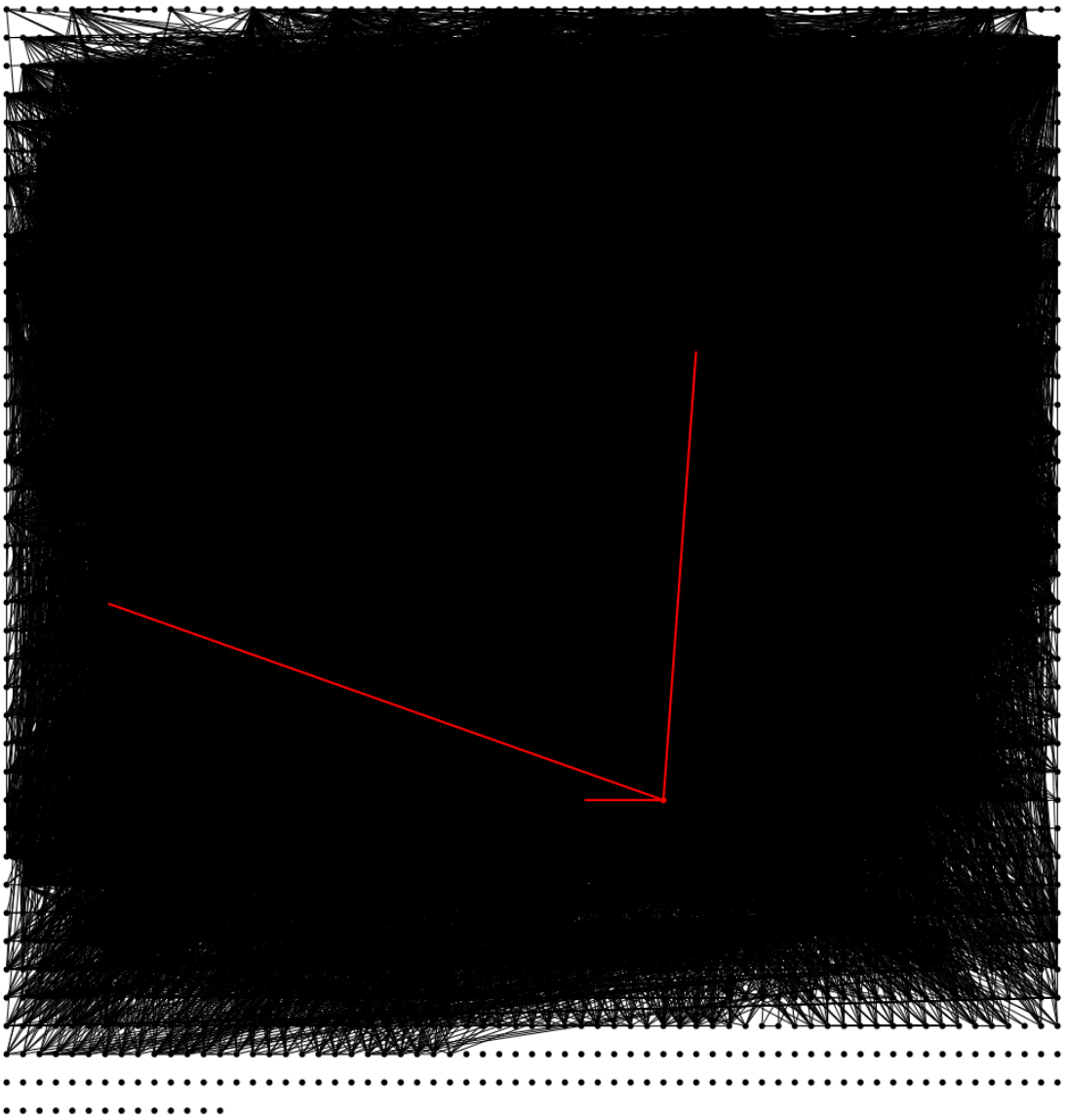
8.1 GRAF SOCIÁLNÍ SÍTĚ

Nejdříve jsem použil tento program pro vizualizaci vygenerovaných dat z fóra, které se nachází na webové adrese: http://www.specnaz.cz/forum. Toto fórum jsem procházel do hloubky zanoření, která odpovídala zobrazení druhé stránky v daném tématu. Toto zanoření znamenalo projít

přes 3500 webových stránek. Při tomto zanoření jsem z vygenerovaného GraphML souboru dostal 2551 jednotlivých uzlů sociální sítě. Při vykreslování tohoto grafu se v některých rozloženích schématu pro vykreslení sociální sítě v notebooku s operační pamětí 4GB a grafickou pamětí 256MB nestačila paměť.

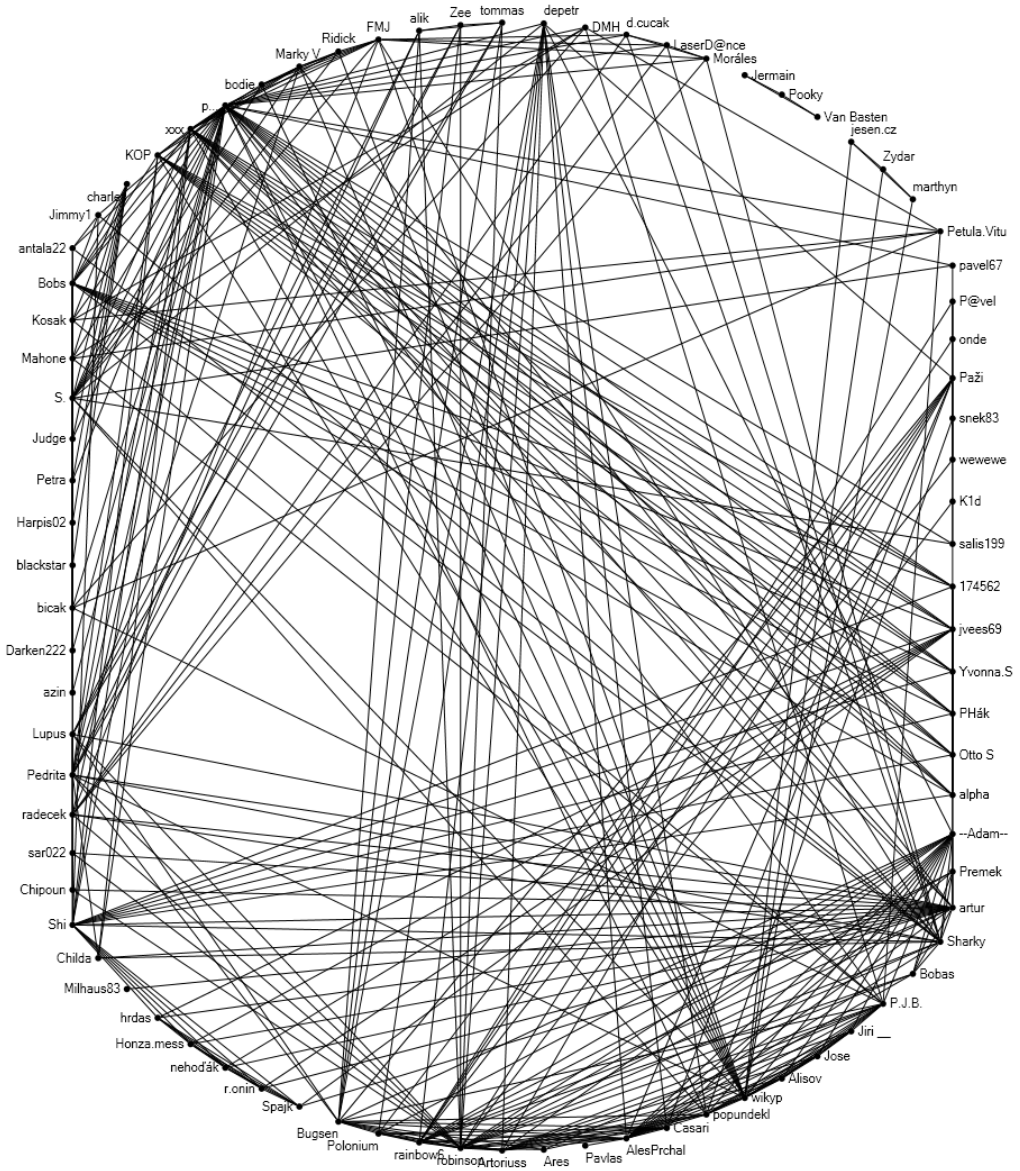
Ve vygenerovaném grafu můžeme vidět, že mezi jednotlivými uzly je spousta hran. Tudíž tyto hrany splývají v jednu plochu. Proto jsem v grafu vyznačil jako ukázku jeden uzel, aby bylo vidět, že hrany mezi uzly existují. U některých uzlů jde vidět izolovanost jednotlivých uzlů, která znamená, že nějaký uživatel vytvořil nějaké téma, ale již mu nikdo nepřispěl žádným příspěvkem.

Obrázek číslo 2: Vygenerovaný graf sociální sítě celého fóra



Jelikož graf na obrázku číslo 2 je nepřehledný a získat z něj informace je obtížné, provedl jsem také analýzu kategorie tohoto fóra. Kategorií jsem vybral podfórum věnující se dopravní policii. Z této kategorie vzniklo 86 jednotlivých uzlů spolu propojených. Tedy žádné téma nezůstalo bez příspěvku. Do vygenerovaného grafu jsem vložil také jména jednotlivých uživatelů, aby bylo vidět, kdo s kým v této kategorii sociální sítě komunikuje.

Obrázek č 3: Vygenerovaný graf sociální sítě kategorie webového fóra



ZÁVĚR

Cílem této práce bylo navrhnout aplikaci, která bude zpracovávat dat z diskusních fór na internetu. Nejdříve jsem ve své práci uvedl základní pojmy týkající se diskusních fór a sociálních sítí. Dále pak jsem na základě literatury uvedl jak z těchto diskusních fór získávat data a jak tyto data zpracovávat.

Při samotném návrhu aplikace jsem se seznámil s funkcí robotů prohledávajících webové stránky. Na základě vědomostí o regulárních výrazech jsem navrhl příklad jak získávat data z těchto stránek. Pomocí těchto znalostí jsem navrhl aplikaci, která získává a zpracovává data z diskusních fór.

Při návrhu jsem vycházel z toho, aby se jednalo o jednoduchou aplikaci, která bude snadno měnitelná bez velkého zásahu programátora do zdrojového kódu. Silnou stránkou této aplikace je, že programátor může pro získání dat jednoduše doimplementovat svého vlastního robota a svůj vlastní parser na webové fórum, se kterým potřebuje pracovat. Další výhodou aplikace je, že je uživatelsky přívětivá a jednoduchá. Uživatel tudíž její ovládání zvládne bez problémů, jelikož aplikace uživatele vede a nedochází tak ke zbytečným chybám.

V poslední části své práce jsem vygeneroval příklady vizualizace analýzy dat z diskusních fór. Jednotlivé vizualizace identifikovali uživatele, kteří mezi sebou na daném fóru komunikovali. Jedna vizualizace byla z celého diskusního fóra, omezená pouze hloubkou zanoření, která odpovídala druhé stránce v daném tématu. Druhá vizualizace byla omezená ještě na podkategorii tohoto webového fóra. Zaměřením na toto podfórum se počet uzlů (účastníků) z původních přes dva a půl tisíce uzlů sítě, zredukoval na 86 a v rámci vygenerovaného grafu již mohla být zobrazena i jména jednotlivých uživatelů internetové diskuse, která se týkala dopravní policie.

V rámci dalšího magisterského studia lze tuto aplikaci dále zdokonalovat. Pro dalšího vylepšení aplikace bych navýšil její rychlost doimplementováním více vláken do této aplikace. Dále jí pak doimplementovat ji na jiná diskusní fóra. Tím získat ještě kvalitnější data pro následnou kompletní analýzu sítě, která by obsahovala již sofistikovanější metody analýzy.

LITERATURA

[1] Hlaváček Lukáš, Výmola Michal, přednáška - Sociální sítě, FEI – Technická univerzita Ostrava

[2] Beránek Ladislav, Síťová analýza v marketingu, In Znalosti 2008.
URL: http://znalosti2008.fiit.stuba.sk/download/articles/znalosti2008-Beranek.pdf
[2010-4-4]

[3] URL:
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.72.4705&rep=rep1&type=pdf
[2010-5-4]

[4] URL: http://graphml.graphdrawing.org/
[2010-5-4]

[5] URL: http://dspace.upce.cz/bitstream/10195/24367/1/
KuklaM_Vyuziti%20regularnich_JR_2007.pdf
[2010-4-30]

[6] Kubíček Michal, Velký průvodce SEO, 1. vydání, nakladatelství Computer Press a.s., Brno, 2008

[7] Bayer Jürgen,C# 2005 – Velká kniha řešení, 1. vydání, nakladatelství Computer Press a.s., Brno, 2007